DETAILED ACTION

1. This Office action is in response to the amendment filed on March 14, 2008.

2. **Claims 42-80** are pending.

3. **Claims 62 and 72** have been amended.

4. **Claims 1-41** have been cancelled.

5. The objection to the title is withdrawn in view of Applicant's amendments to the title.

6. The 35 U.S.C. § 112, second paragraph, rejections of Claims 62-80 are withdrawn in view of Applicant's amendments to the claims.

Response to Amendment

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. **Claims 42-45, 49, 52-55, 59, 62-65, 69, 72-74, and 78** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Eisenstein et al., "Applying Model-Based Techniques to the Development of UIs for Mobile Computers," 2001 (hereinafter "Eisenstein")** in view of **Puerta et al., "Towards a General Computational Framework for Model-Based Interface Development Systems," 1999 (hereinafter "Puerta")**.

As per **Claim 42**, Eisenstein discloses:

- receiving a domain model, a user model, a task model, a device model, and a presentation elements library, wherein the domain model defines application requirements for which the user interface is to be used, wherein the user model defines user requirements of users who are to interface with the user interface, wherein the task model defines task requirements of tasks to be performed between the user interface and users, wherein the device model defines interaction delivery devices that are available to deliver the user interface, and wherein the presentation elements library contains a set of display objects used to present information to or acquire information from a user of the user interface being designed *(see Figure 2; Page 70, "A platform model describes the various computer systems that may run a UI. This model includes information regarding the constraints placed on the UI by the platform. The platform model contains an element for each platform that is supported, and each element contains attributes describing features and constraints." and "A presentation model describes the visual appearance of the user interface. The presentation model includes information describing the hierarchy of windows and their widgets (e.g., sliders, list boxes), stylistic choices, and the selection and placement of these widgets."; Page 71, "A task model is a structured representation of the tasks that the user of the software may want to perform. The task model is hierarchically decomposed into subtasks, and information regarding goals, preconditions, and postconditions may be supplied." and "For many applications, it is essential to model the users themselves, especially when there are multiple users with different preferences, abilities, and privileges. It is also often appropriate to model the domain characteristics of the tasks supported by the UI. Such information often guides the selection of widgets.")*;

- generating a set of presentations, wherein each presentation in the set of presentations comprises an interaction delivery device and a display object that meets a set of requirements of the interaction delivery device, wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the task requirements defined by the task model, and wherein the display object is selected from the set of display objects in the presentation elements library that meets the task requirements defined by the task model and the application requirements defined by the domain model *(see Figures 2 and 3; Page 74, "The designer should create mappings between platforms (or classes of platforms) and tasks (or sets of tasks). Additional mappings are then created between task elements and presentation structures that are optimized for a given set of tasks. We can assume these mappings are transitive; as a result, the appropriate presentation model is associated with each platform, based on mappings through the task model.")*; and

- displaying the set of presentations to a user interface designer *(see Page 72, "Under our proposed architecture, it is still left to the interface designer to specify a set of alternative presentation structures.").*

However, Eisenstein does not disclose:

- wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the user requirements defined by the user model, and wherein the display object is selected from a set of display objects in the presentation elements library that meets the application requirements defined by the domain model.

Puerta discloses:

- wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the user requirements defined by the user model, and wherein the display object is selected from a set of display objects in the presentation elements library that meets the application requirements defined by the domain model *(see Page 173, "Each user may be involved in all tasks in a user-task model, or just in a subset of these tasks. The assignment of users to tasks is a mapping process."; Page 174, "An interface model must also define what objects are involved in the completion of the tasks represented in its user-task model component. Thus it is necessary to map objects to tasks in an interface model.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Puerta into the teaching of Eisenstein to include wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the user requirements defined by the user model, and wherein the display object is selected from a set of display objects in the presentation elements library that meets the application requirements defined by the domain model. The modification would be obvious because one of ordinary skill in the art would be motivated to build and refine the interface model to produce a user interface *(see Puerta – Page 171).*

As per **Claim 43**, the rejection of **Claim 42** is incorporated; and Eisenstein further discloses:

- responsive to at least one input from the user interface designer, generating the user interface *(see Figures 6 and 7).*

As per **Claim 44**, the rejection of **Claim 42** is incorporated; and Eisenstein further discloses:

- wherein generating a set of presentations is performed by a reasoning engine *(see Figure 3; Page 72, "This mediator should determine the maximum usable screen resolution for the relevant device, and evaluate the amount of screen resolution required by each presentation structure alternative. It can then select the presentation structure that consumes an amount of screen resolution that falls just under the maximum (fig. 3).")*.

As per **Claim 45**, the rejection of **Claim 42** is incorporated; and Eisenstein further discloses:

- matching capabilities of the interactive delivery devices in the device model to task requirements defined in the task model *(see Figure 5; Page 74, "The designer should create mappings between platforms (or classes of platforms) and tasks (or sets of tasks). Additional mappings are then created between task elements and presentation structures that are optimized for a given set of tasks. We can assume these mappings are transitive; as a result, the appropriate presentation model is associated with each platform, based on mappings through the task model.")*; and

- matching capabilities of display objects in the presentation elements library to task requirements defined in the task model *(see Figure 5; Page 74, "Additional mappings are then created between task elements and presentation structures that are optimized for a given set of tasks.")*.

However, Eisenstein does not disclose:

- matching capabilities of the interactive delivery devices in the device model to user requirements defined in the user model; and

- matching capabilities of display objects in the presentation elements library to application requirements defined in the domain model.

Puerta discloses:

- matching capabilities of the interactive delivery devices in the device model to user requirements defined in the user model *(see Page 173, "Each user may be involved in all tasks in a user-task model, or just in a subset of these tasks. The assignment of users to tasks is a mapping process.")*; and

- matching capabilities of display objects in the presentation elements library to application requirements defined in the domain model *(see Page 174, "An interface model must also define what objects are involved in the completion of the tasks represented in its user-task model component. Thus it is necessary to map objects to tasks in an interface model.")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Puerta into the teaching of Eisenstein to include matching capabilities of the interactive delivery devices in the device model to user requirements defined in the user model; and matching capabilities of display objects in the presentation elements library to application requirements defined in the domain model. The modification would be obvious because one of ordinary skill in the art would be motivated to build and refine the interface model to produce a user interface *(see Puerta – Page 171)*.

As per **Claim 49**, the rejection of **Claim 42** is incorporated; and Eisenstein further discloses:

- wherein the domain model, the user model, the task model, and the device model are expressed in a common notation format *(see Page 70, "The MIMIC modeling language meets these criteria, and it is the language we have chosen to use for UI modeling.").*

As per **Claim 52**, Eisenstein discloses:

- creating a domain model, a user model, a task model, a device model, and a presentation elements library, wherein the domain model defines application requirements for which the user interface is to be used, wherein the user model defines user requirements of users who are to interface with the user interface, wherein the task model defines task requirements of tasks to be performed between the user interface and users, wherein the device model defines interaction delivery devices that are available to deliver the user interface, and wherein the presentation elements library contains a set of display objects used to present information to or acquire information from a user of the user interface being designed *(see Figure 2; Page 70, "A platform model describes the various computer systems that may run a UI. This model includes information regarding the constraints placed on the UI by the platform. The platform model contains an element for each platform that is supported, and each element contains attributes describing features and constraints." and "A presentation model describes the visual appearance of the user interface. The presentation model includes information describing the hierarchy of windows and their widgets (e.g., sliders, list boxes), stylistic choices, and the selection and placement of these widgets."; Page 71, "A task model is a structured*

representation of the tasks that the user of the software may want to perform. The task model is hierarchically decomposed into subtasks, and information regarding goals, preconditions, and postconditions may be supplied." and "For many applications, it is essential to model the users themselves, especially when there are multiple users with different preferences, abilities, and privileges. It is also often appropriate to model the domain characteristics of the tasks supported by the UI. Such information often guides the selection of widgets.");

- generating a set of presentations, wherein each presentation in the set of presentations comprises an interaction delivery device and a display object that meets a set of requirements of the interaction delivery device, wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the task requirements defined by the task model, and wherein the display object is selected from the set of display objects in the presentation elements library that meets the task requirements defined by the task model and the application requirements defined by the domain model *(see Figures 2 and 3; Page 74, "The designer should create mappings between platforms (or classes of platforms) and tasks (or sets of tasks). Additional mappings are then created between task elements and presentation structures that are optimized for a given set of tasks. We can assume these mappings are transitive; as a result, the appropriate presentation model is associated with each platform, based on mappings through the task model.")*; and

- displaying the set of presentations to a user interface designer *(see Page 72, "Under our proposed architecture, it is still left to the interface designer to specify a set of alternative presentation structures.").*

However, Eisenstein does not disclose:

- storing the domain model, user model, task model, device model, and presentation elements library into computer readable media; and

- wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the user requirements defined by the user model, and wherein the display object is selected from a set of display objects in the presentation elements library that meets the application requirements defined by the domain model.

Official Notice is taken that it is old and well-known within the computing art to store a computer program or components of the computer program in a computer readable media. In a computing system, components of a computer program are stored in a computer readable media so a processing unit may execute the instructions stored therein. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include storing the domain model, user model, task model, device model, and presentation elements library into computer readable media. The modification would be obvious because one of ordinary skill in the art would be motivated to execute the components of the computer program.

Puerta discloses:

- wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the user requirements defined by the user model, and wherein the display object is selected from a set of display objects in the presentation elements library that meets the application requirements defined by the domain model *(see Page 173, "Each user may be involved in all tasks in a user-task model, or just in a subset of these tasks. The assignment of users to tasks is a mapping process."; Page 174, "An interface model must*

also define what objects are involved in the completion of the tasks represented in its user-task model component. Thus it is necessary to map objects to tasks in an interface model.").

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Puerta into the teaching of Eisenstein to include wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the user requirements defined by the user model, and wherein the display object is selected from a set of display objects in the presentation elements library that meets the application requirements defined by the domain model. The modification would be obvious because one of ordinary skill in the art would be motivated to build and refine the interface model to produce a user interface *(see Puerta – Page 171).*

As per **Claim 53**, the rejection of **Claim 52** is incorporated; and Eisenstein further discloses:

- responsive to at least one input from the user interface designer, generating the user interface *(see Figures 6 and 7).*

As per **Claim 54**, the rejection of **Claim 52** is incorporated; and Eisenstein further discloses:

- wherein generating a set of presentations is performed by a reasoning engine *(see Figure 3; Page 72, "This mediator should determine the maximum usable screen resolution for the relevant device, and evaluate the amount of screen resolution required by each presentation*

structure alternative. It can then select the presentation structure that consumes an amount of screen resolution that falls just under the maximum (fig. 3).").

As per **Claim 55**, the rejection of **Claim 52** is incorporated; and Eisenstein further discloses:

- matching capabilities of the interactive delivery devices in the device model to task requirements defined in the task model *(see Figure 5; Page 74, "The designer should create mappings between platforms (or classes of platforms) and tasks (or sets of tasks). Additional mappings are then created between task elements and presentation structures that are optimized for a given set of tasks. We can assume these mappings are transitive; as a result, the appropriate presentation model is associated with each platform, based on mappings through the task model.")*; and

- matching capabilities of display objects in the presentation elements library to task requirements defined in the task model *(see Figure 5; Page 74, "Additional mappings are then created between task elements and presentation structures that are optimized for a given set of tasks.").*

However, Eisenstein does not disclose:

- matching capabilities of the interactive delivery devices in the device model to user requirements defined in the user model; and

- matching capabilities of display objects in the presentation elements library to application requirements defined in the domain model.

Puerta discloses:

- matching capabilities of the interactive delivery devices in the device model to user requirements defined in the user model *(see Page 173, "Each user may be involved in all tasks in a user-task model, or just in a subset of these tasks. The assignment of users to tasks is a mapping process.")*; and

- matching capabilities of display objects in the presentation elements library to application requirements defined in the domain model *(see Page 174, "An interface model must also define what objects are involved in the completion of the tasks represented in its user-task model component. Thus it is necessary to map objects to tasks in an interface model.")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Puerta into the teaching of Eisenstein to include matching capabilities of the interactive delivery devices in the device model to user requirements defined in the user model; and matching capabilities of display objects in the presentation elements library to application requirements defined in the domain model. The modification would be obvious because one of ordinary skill in the art would be motivated to build and refine the interface model to produce a user interface *(see Puerta – Page 171)*.

As per **Claim 59**, the rejection of **Claim 52** is incorporated; and Eisenstein further discloses:

- wherein the domain model, the user model, the task model, and the device model are expressed in a common notation format *(see Page 70, "The MIMIC modeling language meets these criteria, and it is the language we have chosen to use for UI modeling.")*.

As per **Claim 62**, Eisenstein discloses:

- wherein the domain model defines application requirements for which the user interface is to be used *(see Page 71, "It is also often appropriate to model the domain characteristics of the tasks supported by the UI. Such information often guides the selection of widgets.")*;

- wherein the user model defines user requirements of users who are to interface with the user interface *(see Page 71, "For many applications, it is essential to model the users themselves, especially when there are multiple users with different preferences, abilities, and privileges.")*;

- wherein the task model defines task requirements of tasks to be performed between the user interface and users who are to interface with the user interface *(see Page 71, "A task model is a structured representation of the tasks that the user of the software may want to perform. The task model is hierarchically decomposed into subtasks, and information regarding goals, preconditions, and postconditions may be supplied.")*;

- wherein the device model defines interaction delivery devices that are available to deliver the user interface *(see Page 70, "A platform model describes the various computer systems that may run a UI. This model includes information regarding the constraints placed on the UI by the platform. The platform model contains an element for each platform that is supported, and each element contains attributes describing features and constraints.")*;

- wherein the presentation elements library contains a set of display objects used to present information to or acquire information from a user of the user interface being designed *(see Figure 2; Page 70, "A presentation model describes the visual appearance of the user*

interface. The presentation model includes information describing the hierarchy of windows and their widgets (e.g., sliders, list boxes), stylistic choices, and the selection and placement of these widgets.");

- generating a set of presentations, wherein each presentation in the set of presentations comprises an interaction delivery device and a display object that meets a set of requirements of the interaction delivery device, wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the task requirements defined by the task model, and wherein the display object is selected from the set of display objects in the presentation elements library that meets the task requirements defined by the task model and the application requirements defined by the domain model *(see Figures 2 and 3; Page 74, "The designer should create mappings between platforms (or classes of platforms) and tasks (or sets of tasks). Additional mappings are then created between task elements and presentation structures that are optimized for a given set of tasks. We can assume these mappings are transitive; as a result, the appropriate presentation model is associated with each platform, based on mappings through the task model.")*; and

- displaying the set of presentations to a user interface designer *(see Page 72, "Under our proposed architecture, it is still left to the interface designer to specify a set of alternative presentation structures.").*

However, Eisenstein does not disclose:

- storing a domain model, a user model, a task model, a device model, and a presentation elements library into computer readable media; and

- wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the user requirements defined by the user model, and wherein the display object is selected from a set of display objects in the presentation elements library that meets the application requirements defined by the domain model.

Official Notice is taken that it is old and well-known within the computing art to store a computer program or components of the computer program in a computer readable media. In a computing system, components of a computer program are stored in a computer readable media so a processing unit may execute the instructions stored therein. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include storing a domain model, a user model, a task model, a device model, and a presentation elements library into computer readable media. The modification would be obvious because one of ordinary skill in the art would be motivated to execute the components of the computer program.

Puerta discloses:

- wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the user requirements defined by the user model, and wherein the display object is selected from a set of display objects in the presentation elements library that meets the application requirements defined by the domain model *(see Page 173, "Each user may be involved in all tasks in a user-task model, or just in a subset of these tasks. The assignment of users to tasks is a mapping process."; Page 174, "An interface model must also define what objects are involved in the completion of the tasks represented in its user-task model component. Thus it is necessary to map objects to tasks in an interface model.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Puerta into the teaching of Eisenstein to include wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the user requirements defined by the user model, and wherein the display object is selected from a set of display objects in the presentation elements library that meets the application requirements defined by the domain model. The modification would be obvious because one of ordinary skill in the art would be motivated to build and refine the interface model to produce a user interface *(see Puerta – Page 171)*.

As per **Claim 63**, the rejection of **Claim 62** is incorporated; and Eisenstein further discloses:

- responsive to at least one input from the user interface designer, generating the user interface *(see Figures 6 and 7)*.

As per **Claim 64**, the rejection of **Claim 62** is incorporated; and Eisenstein further discloses:

- wherein generating a set of presentations is performed by a reasoning engine *(see Figure 3; Page 72, "This mediator should determine the maximum usable screen resolution for the relevant device, and evaluate the amount of screen resolution required by each presentation structure alternative. It can then select the presentation structure that consumes an amount of screen resolution that falls just under the maximum (fig. 3).")*.

As per **Claim 65**, the rejection of **Claim 62** is incorporated; and Eisenstein further discloses:

- matching capabilities of the interactive delivery devices in the device model to task requirements defined in the task model *(see Figure 5; Page 74, "The designer should create mappings between platforms (or classes of platforms) and tasks (or sets of tasks). Additional mappings are then created between task elements and presentation structures that are optimized for a given set of tasks. We can assume these mappings are transitive; as a result, the appropriate presentation model is associated with each platform, based on mappings through the task model.")*; and

- matching capabilities of display objects in the presentation elements library to task requirements defined in the task model *(see Figure 5; Page 74, "Additional mappings are then created between task elements and presentation structures that are optimized for a given set of tasks.").*

However, Eisenstein does not disclose:

- matching capabilities of the interactive delivery devices in the device model to user requirements defined in the user model; and

- matching capabilities of display objects in the presentation elements library to application requirements defined in the domain model.

Puerta discloses:

- matching capabilities of the interactive delivery devices in the device model to user requirements defined in the user model *(see Page 173, "Each user may be involved in all tasks in*

a user-task model, or just in a subset of these tasks. The assignment of users to tasks is a mapping process."); and

- matching capabilities of display objects in the presentation elements library to application requirements defined in the domain model *(see Page 174, "An interface model must also define what objects are involved in the completion of the tasks represented in its user-task model component. Thus it is necessary to map objects to tasks in an interface model.")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Puerta into the teaching of Eisenstein to include matching capabilities of the interactive delivery devices in the device model to user requirements defined in the user model; and matching capabilities of display objects in the presentation elements library to application requirements defined in the domain model. The modification would be obvious because one of ordinary skill in the art would be motivated to build and refine the interface model to produce a user interface *(see Puerta – Page 171)*.

As per **Claim 69**, the rejection of **Claim 62** is incorporated; and Eisenstein further discloses:

- wherein the domain model, the user model, the task model, and the device model are expressed in a common notation format *(see Page 70, "The MIMIC modeling language meets these criteria, and it is the language we have chosen to use for UI modeling.")*.

Claims 72-74 and 78 are computer readable medium claims corresponding to the method claims above (Claims 42, 43, 45, and 49) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 42, 43, 45, and 49.

9. **Claims 46-48, 56-58, 66-68, and 75-77** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Eisenstein** in view of **Puerta** as applied to Claims 42, 52, 62, and 72 above, and further in view of **US 6,243,713 (hereinafter "Nelson")**.

As per **Claim 46**, the rejection of **Claim 42** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein generating a set of presentations further comprises scoring each presentation based at least in part on the application requirements defined in the domain model, the user requirements defined in the user model, and the task requirements defined in the task model.

Nelson discloses:

- wherein generating a set of presentations further comprises scoring each presentation based at least in part on the application requirements defined in the domain model, the user requirements defined in the user model, and the task requirements defined in the task model *(see Column 26: 44-48, "Once all candidate documents are scored, the final scores are sorted, and the documents presented to the user, providing the best scoring documents first. A threshold may be used to select a limited number of the best scoring documents if desired.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Nelson into the teaching of Eisenstein to

include wherein generating a set of presentations further comprises scoring each presentation based at least in part on the application requirements defined in the domain model, the user requirements defined in the user model, and the task requirements defined in the task model. The modification would be obvious because one of ordinary skill in the art would be motivated to enhance usability.

As per **Claim 47**, the rejection of **Claim 46** is incorporated; however, Eisenstein and Puerta do not disclose:

- sorting each presentation according to its score.

Nelson discloses:

- sorting each presentation according to its score *(see Column 26: 44-48, "Once all candidate documents are scored, the final scores are sorted, and the documents presented to the user, providing the best scoring documents first. A threshold may be used to select a limited number of the best scoring documents if desired.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Nelson into the teaching of Eisenstein to include sorting each presentation according to its score. The modification would be obvious because one of ordinary skill in the art would be motivated to enhance usability.

As per **Claim 48**, the rejection of **Claim 42** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein displaying the set of presentations to a user interface designer further comprises displaying each presentation in a ranked list according to score.

Nelson discloses:

- wherein displaying the set of presentations to a user interface designer further comprises displaying each presentation in a ranked list according to score *(see Column 26: 44-48, "Once all candidate documents are scored, the final scores are sorted, and the documents presented to the user, providing the best scoring documents first. A threshold may be used to select a limited number of the best scoring documents if desired.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Nelson into the teaching of Eisenstein to include wherein displaying the set of presentations to a user interface designer further comprises displaying each presentation in a ranked list according to score. The modification would be obvious because one of ordinary skill in the art would be motivated to enhance usability.

As per **Claim 56**, the rejection of **Claim 52** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein generating a set of presentations further comprises scoring each presentation based at least in part on the application requirements defined in the domain model, the user requirements defined in the user model, and the task requirements defined in the task model.

Nelson discloses:

- wherein generating a set of presentations further comprises scoring each presentation based at least in part on the application requirements defined in the domain model, the user

requirements defined in the user model, and the task requirements defined in the task model *(see Column 26: 44-48, "Once all candidate documents are scored, the final scores are sorted, and the documents presented to the user, providing the best scoring documents first. A threshold may be used to select a limited number of the best scoring documents if desired.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Nelson into the teaching of Eisenstein to include wherein generating a set of presentations further comprises scoring each presentation based at least in part on the application requirements defined in the domain model, the user requirements defined in the user model, and the task requirements defined in the task model. The modification would be obvious because one of ordinary skill in the art would be motivated to enhance usability.

As per **Claim 57**, the rejection of **Claim 56** is incorporated; however, Eisenstein and Puerta do not disclose:

- sorting each presentation according to its score.

Nelson discloses:

- sorting each presentation according to its score *(see Column 26: 44-48, "Once all candidate documents are scored, the final scores are sorted, and the documents presented to the user, providing the best scoring documents first. A threshold may be used to select a limited number of the best scoring documents if desired.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Nelson into the teaching of Eisenstein to

include sorting each presentation according to its score. The modification would be obvious because one of ordinary skill in the art would be motivated to enhance usability.

As per **Claim 58**, the rejection of **Claim 52** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein displaying the set of presentations to a user interface designer further comprises displaying each presentation in a ranked list according to score.

Nelson discloses:

- wherein displaying the set of presentations to a user interface designer further comprises displaying each presentation in a ranked list according to score *(see Column 26: 44-48, "Once all candidate documents are scored, the final scores are sorted, and the documents presented to the user, providing the best scoring documents first. A threshold may be used to select a limited number of the best scoring documents if desired.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Nelson into the teaching of Eisenstein to include wherein displaying the set of presentations to a user interface designer further comprises displaying each presentation in a ranked list according to score. The modification would be obvious because one of ordinary skill in the art would be motivated to enhance usability.

As per **Claim 66**, the rejection of **Claim 62** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein generating a set of presentations further comprises scoring each presentation based at least in part on the application requirements defined in the domain model, the user requirements defined in the user model, and the task requirements defined in the task model.

Nelson discloses:

- wherein generating a set of presentations further comprises scoring each presentation based at least in part on the application requirements defined in the domain model, the user requirements defined in the user model, and the task requirements defined in the task model *(see Column 26: 44-48, "Once all candidate documents are scored, the final scores are sorted, and the documents presented to the user, providing the best scoring documents first. A threshold may be used to select a limited number of the best scoring documents if desired.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Nelson into the teaching of Eisenstein to include wherein generating a set of presentations further comprises scoring each presentation based at least in part on the application requirements defined in the domain model, the user requirements defined in the user model, and the task requirements defined in the task model. The modification would be obvious because one of ordinary skill in the art would be motivated to enhance usability.

As per **Claim 67**, the rejection of **Claim 66** is incorporated; however, Eisenstein and Puerta do not disclose:

- sorting each presentation according to its score.

Nelson discloses:

- sorting each presentation according to its score *(see Column 26: 44-48, "Once all candidate documents are scored, the final scores are sorted, and the documents presented to the user, providing the best scoring documents first. A threshold may be used to select a limited number of the best scoring documents if desired.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Nelson into the teaching of Eisenstein to include sorting each presentation according to its score. The modification would be obvious because one of ordinary skill in the art would be motivated to enhance usability.

As per **Claim 68**, the rejection of **Claim 62** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein displaying the set of presentations to a user interface designer further comprises displaying each presentation in a ranked list according to score.

Nelson discloses:

- wherein displaying the set of presentations to a user interface designer further comprises displaying each presentation in a ranked list according to score *(see Column 26: 44-48, "Once all candidate documents are scored, the final scores are sorted, and the documents presented to the user, providing the best scoring documents first. A threshold may be used to select a limited number of the best scoring documents if desired.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Nelson into the teaching of Eisenstein to include wherein displaying the set of presentations to a user interface designer further comprises

displaying each presentation in a ranked list according to score. The modification would be obvious because one of ordinary skill in the art would be motivated to enhance usability.

Claims 75-77 are rejected for the same reasons set forth in the rejections of Claims 46-48.

10. **Claims 50, 60, 70, and 79** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Eisenstein** in view of **Puerta** as applied to Claims 49, 59, 69, and 78 above, and further in view of **"Resource Description Framework (RDF) Model and Syntax," 1997 (hereinafter "RDF1997")**.

As per **Claim 50**, the rejection of **Claim 49** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein the common notation format adheres to the Resource Description Framework specification.

RDF1997 discloses:

- wherein the common notation format adheres to the Resource Description Framework specification *(see Section 1, "RDF metadata can be used in a variety of application areas ...")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of RDF1997 into the teaching of Eisenstein to include wherein the common notation format adheres to the Resource Description Framework specification. The modification would be obvious because one of ordinary skill in the art would

be motivated to provide interoperability between applications that exchange machine understandable information on the Web *(see RDF1997 – Section 1)*.

As per **Claim 60**, the rejection of **Claim 59** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein the common notation format adheres to the Resource Description Framework specification.

RDF1997 discloses:

- wherein the common notation format adheres to the Resource Description Framework specification *(see Section 1, "RDF metadata can be used in a variety of application areas ...")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of RDF1997 into the teaching of Eisenstein to include wherein the common notation format adheres to the Resource Description Framework specification. The modification would be obvious because one of ordinary skill in the art would be motivated to provide interoperability between applications that exchange machine understandable information on the Web *(see RDF1997 – Section 1)*.

As per **Claim 70**, the rejection of **Claim 69** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein the common notation format adheres to the Resource Description Framework specification.

RDF1997 discloses:

- wherein the common notation format adheres to the Resource Description Framework specification *(see Section 1, "RDF metadata can be used in a variety of application areas ...").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of RDF1997 into the teaching of Eisenstein to include wherein the common notation format adheres to the Resource Description Framework specification. The modification would be obvious because one of ordinary skill in the art would be motivated to provide interoperability between applications that exchange machine understandable information on the Web *(see RDF1997 – Section 1).*

Claim 79 is rejected for the same reason set forth in the rejection of Claim 50.

11. **Claims 51, 61, 71, and 80** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Eisenstein** in view of **Puerta** as applied to Claims 42, 52, 62, and 72 above, and further in view of **"Extensible Markup Language (XML) 1.0," 1998 (hereinafter "XML1998").**

As per **Claim 51**, the rejection of **Claim 42** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein each presentation is an XML file.

XML1998 discloses:

- wherein each presentation is an XML file *(see Section 1, "Extensible Markup Language, abbreviated XML, describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of XML1998 into the teaching of Eisenstein to include wherein each presentation is an XML file. The modification would be obvious because one of ordinary skill in the art would be motivated to support a wide variety of applications *(see XML1998 – Section 1.1)*.

As per **Claim 61**, the rejection of **Claim 52** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein each presentation is an XML file.

XML1998 discloses:

- wherein each presentation is an XML file *(see Section 1, "Extensible Markup Language, abbreviated XML, describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them.")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of XML1998 into the teaching of Eisenstein to include wherein each presentation is an XML file. The modification would be obvious because one of ordinary skill in the art would be motivated to support a wide variety of applications *(see XML1998 – Section 1.1)*.

As per **Claim 71**, the rejection of **Claim 62** is incorporated; however, Eisenstein and Puerta do not disclose:

- wherein each presentation is an XML file.

XML1998 discloses:

- wherein each presentation is an XML file *(see Section 1, "Extensible Markup Language, abbreviated XML, describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of XML1998 into the teaching of Eisenstein to include wherein each presentation is an XML file. The modification would be obvious because one of ordinary skill in the art would be motivated to support a wide variety of applications *(see XML1998 – Section 1.1).*

Claim 80 is rejected for the same reason set forth in the rejection of Claim 51.

Response to Arguments

12. Applicant's arguments filed on March 14, 2008 have been fully considered, but they are not persuasive.

In the Remarks, Applicant argues:

a) Generally, interaction delivery devices are devices that can be used to deliver user interfaces to a user. (Specification at page 24, lines 6-9) For example, in the prescription drug store domain described in the Specification, interaction delivery devices may include web browsers, PDAs, telephonic interaction delivery devices, etc. (Specification at page 24, lines 3-6)

As set forth in claim 42, device models define the interaction delivery devices that are available to deliver the interaction delivery device.

Puerta simply does not discuss device models, and therefore, does not teach selecting an interaction delivery device from a set of interaction delivery devices in a device model. The Examiner asserted that Puerta, in discussing a "task model" and "map[ping] objects to tasks in an interface model," teaches the limitation of selecting an interaction delivery device from a set of interaction delivery devices in a device model. (Office Action at pages 6-7) The cited passage only deals with task models and the mapping of objects to tasks in an interface model (i.e. a user interface model), and does not teach anything about device models. The remainder of Puerta is equally lacking as the "basic components" of Puerta's interface model "are the user-task model, the user model, the domain, the presentation model, and the dialog model." (Puerta at column 3) A presentation model "is a representation of the visual haptic and auditory elements that a user interface offers to its users" and a dialog model "defines the way in which the presentation model interacts with the user." Id. Thus, none of the components of Puerta resemble a device model. Without any discussion of device models, Puerta does not teach selecting an interaction delivery device from a set of interaction delivery devices in a device model.

Examiner's response:

a) Examiner disagrees. Applicant's arguments are not persuasive for at least the following reasons:

First, with respect to Applicant's attempt to define "interaction delivery device" by referring to examples from the specification, although the claims are interpreted in light of the

specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Second, the claimed invention is directed to generating a set of presentations for a user interface by mapping a domain model, a user model, a task model, a device model, and a presentation library. Eisenstein's invention is directed to generating a set of presentations for a user interface by mapping a task model, a platform model (a device model), and a presentation model (a presentation library). Eisenstein further discloses that these are not the only models that are relevant to the development of UIs for mobile computers, a user model and a domain model are essential as well *(see Page 71, "For many applications, it is essential to model the users themselves, especially when there are multiple users with different preferences, abilities, and privileges. It is also often appropriate to model the domain characteristics of the tasks supported by the UI. Such information often guides the selection of widgets.")*. Puerta's invention is directed to generating a set of presentations for a user interface by mapping a user-task model, a domain model, a user model, a presentation model, and a dialog model. Examiner would like to emphasize that Jacob Eisenstein and Angel Puerta are the authors of the two cited prior art. The two cited prior art are parts of the authors' continuing research in model-based development of UIs for mobile computers. Thus, one of ordinary skill in the art would be motivated to combine the teachings of the two cited prior art.

As previously pointed out in the Non-Final Rejection (mailed on 12/20/2007), Eisenstein does not disclose the limitation of "wherein the interaction delivery device is selected from a set of interaction delivery devices in the device model that meets the user requirements defined by the user model, and wherein the display object is selected from a set of display objects in the

presentation elements library that meets the application requirements defined by the domain model (emphasis added)." In other words, Eisenstein does not disclose a mapping between the device model and the user model and a mapping between the presentation library and the domain model. Note that, however, Eisenstein discloses a mapping between the platform model and the task model *(see Page 74, "The designer should create mappings between platforms (or classes of platforms) and tasks (or sets of tasks).")* and a mapping between the task model and the presentation model *(see Page 74, "Additional mappings are then created between task elements and presentation structures that are optimized for a given set of tasks.")*. Puerta discloses a mapping between the task model and the user model *(see Page 173, "Task-User Mappings")* and a mapping between the task model and the domain model *(see Page 174, "Task-Domain Mappings")*. Eisenstein also discloses that these mappings are transitive *(see Page 74, "We can assume these mappings are transitive; as a result, the appropriate presentation model is associated with each platform, based on mappings through the task model.")*, meaning that if a mapping exists between a first model and a second model and a mapping exists between the second model and a third model, this would imply that a mapping exists between the first model and the third model. Thus, when combined, a mapping exists between the platform model and the user model via the task model and a mapping also exists between the presentation model and the domain model via the task model.

For at least the reasons set forth above, the rejections made under 35 U.S.C. 103(a) over the prior art of record with regard to Claims 42, 52, 62, and 72 are proper and therefore, maintained.

Conclusion

13. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/QC/
May 31, 2008

/Wei Zhen/

Supervisory Patent Examiner, Art Unit 2191